Computation of cubical homology, cohomology, and (co)homological operations via chain contraction

Paweł Pilarczyk · Pedro Real

Abstract We introduce algorithms for the computation of homology, cohomology, and related operations on cubical cell complexes, using the technique based on a chain contraction from the original chain complex to a reduced one that represents its homology. This work is based on previous results for simplicial complexes, and uses Serre's diagonalization for cubical cells. An implementation in C++ of the intro-duced algorithms is available at http://www.pawelpilarczyk.com/chaincon/ together with some examples. The paper is self-contained as much as possible, and is written at a very elementary level, so that basic knowledge of algebraic topology should be sufficient to follow it.

Keywords Algorithm · Software · Homology · Cohomology · Computational homology · Cup product · Alexander-Whitney coproduct · Chain homotopy · Chain contraction · Cubical complex

Mathematics Subject Classification (2010) 55N35 · 52B99 · 55U15 · 55U30 · 55-04

Communicated by: D. N. Arnold

P. Pilarczyk (✉)
Universidade do Minho, Centro de Matemática, Campus de Gualtar, 4710-057 Braga, Portugal
e-mail: pawel.pilarczyk@ist.ac.at
URL: http://www.pawelpilarczyk.com/

P. Real
Departamento de Matemática Aplicada I, E. T. S. de Ingeniería Informática, Universidad de Sevilla, Avenida Reina Mercedes s/n, 41012 Sevilla, Spain
e-mail: real@us.es
URL: http://investigacion.us.es/sisius/sis showpub.php?idpers=1021

Present Address:
P. Pilarczyk
IST Austria, Am Campus 1, 3400 Klosterneuburg, Austria

1 Introduction

A *full cubical set* in $\mathbb{R}^n$ is a finite union of n-dimensional boxes of fixed size (called *cubes* for short) aligned with a uniform rectangular grid in $\mathbb{R}^n$. Due to the product structure and alignment with coordinate axes, using full cubical sets for approximating bounded subsets of $\mathbb{R}^n$ is very natural: a cube containing a point $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$ can be instantly calculated by simply truncating the Cartesian coordinates of x down to the nearest grid thresholds. For simplicity of notation, these thresholds can be set to the integers, so that the cubes are of unitary size. Such cubical sets naturally correspond to 2D and 3D binary images.

By analogy to simplicial complexes (see e.g. [44]), sets of cubes with their vertices, edges, faces, etc., yield a natural chain complex structure, which can be used to compute their homology groups. We refer to [34] for a comprehensive study of this subject, and to the [9] and the [10] for a representative implementation of homology computation algorithms focused specifically on cubical sets.

Homology computation of cubical sets has already found some interesting applications. To mention a few of them, homology was used to extract topological information from medical images (e.g. [45]), to classify the complexity of patterns coming from numerical simulations of PDEs (e.g. [20]) or from physical experiments (e.g. [37]), and also in an automatized method for the computation of the Conley index from index pairs constructed as cubical sets in $\mathbb{R}^n$ [41, 48], used e.g. in a method for editing vector fields and extraction of periodic orbits [8], and also very helpful in an automatic method for classification of dynamics in multi-parameter systems [4, 7], which found applications e.g. in population biology [38] and in physics of plasmas [49]. Homology has also been proposed as a reliable criterion for thinning binary images of arbitrary dimension [46].

With the increasing use of data structures based upon cubical sets in various applications, which often require the determination of certain topological features of the sets under consideration, the importance of the development of efficient algorithms for the computation of comprehensive algebraic-topological invariants of such sets is undeniable. Our work is aimed at contributing to this field of research by providing effective algorithms for the computation of comprehensive homological information on cubical sets. A prototype software implementation of these algorithms which allows for experimenting with sample datasets is published at the project's website [47].

In a typical approach to homology computation (see e.g. [1, 44, 60]), which we call the *differential approach* here, the matrices of the boundary homomorphisms ∂_q (the differential of the chain complex) are reduced to the Smith Normal Form (SNF), from which the homology groups are determined. In the *integral approach*, on the other hand, in addition to the computation of the homology groups, one also constructs degree $+1$ homomorphisms $\phi_q \colon C_q \to C_{q+1}$, which record the information on a chain contraction of the entire chain complex C to a reduced chain complex that represents its homology (see e.g. [17, 55, 57]). This contraction provides much more comprehensive homological information which may be important for various applications. In particular, it allows one to calculate a representative cycle for each homology class, to find the homology class of every cycle, and to compute

the coboundary of every homologically trivial cycle (that is, find the chain c' such that $c = \partial c'$ if $[c] = [0]$). Obviously, the computational cost of the integral approach is higher, especially because much more information needs to be stored. This additional information which is normally lost in the differential approach, however, may be essential in specific applications, e.g. for the computation of various homological or cohomological operations, as we show in Section 5.

Intuitively speaking, the profound difference between the differential approach and the integral one is that the former aims at *reducing* the topological information to a minimal linear system that describes the degree of connectivity of the subdivided objects, while the latter provides a dense algebraic skeleton for *representing* these objects. Unfortunately, this latter approach is very under-represented in the literature; only some works of Sergeraert relating to effective homology (see e.g. [54]) and those regarding AT models and AM models (referred to below) emphasize this aspect of homological constructions.

The integral approach was applied to the computation of the cohomology ring and some other cohomology operations in the context of simplicial complexes in [21–24, 26, 50]. The philosophy behind this approach, as described in [23], is to deduce cohomology operation formulas at (co)chain level in terms of face operators on simplices, and to use the chain contractions to correctly transfer these operations to the cohomology level. In order to use this methodology in the cubical setting, the most straightforward idea would be to construct a simplicial representation of a cubical set and to apply the existing theory and algorithms to this simplicial complex. However, this induces a considerable overhead, especially in higher dimensions, not to mention the additional effort necessary to express the obtained simplicial chains in terms of cubical chains. A much more effective solution is thus to transfer the computational cohomology approach from simplicial to cubical setting. Therefore, the main aim of our work is to develop efficient (co)homology algorithms directly on the cubical structure, skipping the intermediate steps. As the main tool for this purpose, we are going to use an algebraic-topological model (AT model, see Definition 1) and an algebraic minimal model (AM model, see Definition 2). These notions have already proved useful in several publications (e.g. [6, 21, 24, 25, 42, 53]).

The worst-case complexity of homology computation, independent of the approach, is in general cubical if the ring of coefficients is a field, or a little higher yet polynomial otherwise, because of the need to compute the SNF of the boundary matrices (either explicitly or implicitly). Although this is too expensive for processing large datasets that appear in practical applications, various reduction techniques help decrease the size of the data considerably without loss of homological information (see e.g. [43]), and thus achieve much better performance in practice. These reduction techniques may come from geometric representation of the data, or be derived from other heuristics. As an interesting example, a physics inspired algorithm for the computation of the first cohomology group on three-dimensional complexes was introduced in [14] whose average running time was linear in their experiments, even though theoretically estimated worst-case complexity was cubical.

Although several software packages that aim at (co)homology computation already exist, none of them, as far as we are aware, can be used to achieve the functionality of our approach. To name a few most prominent examples, Plex and

Dionysus are powerful projects aimed at simplicial persistent (co)homology computation, CHomP and CAPD/RedHom focus on the development of efficient homology algorithms for cubical sets using the differential approach. There are also some modules in large mathematical software packages, such as GAP or SAGE, but they all have limited capabilities and are using the differential approach. In particular, our contribution fills an important gap in the collection of publicly available homological software.

Since the construction of a chain contraction has a considerable additional cost in comparison to using the differential approach (in which all this information is lost), it is obviously pointless to use the integral approach if one aims at the computation of (co)homology groups (possibly with their generators) of a cellular complex alone. The integral approach, however, provides a considerably more efficient solution in cases where the chain contraction is actually useful. For instance, if one needs to compare homology classes of various cycles that are not known a priori then using the differential approach might incur considerable cost of processing each cycle, while the integral approach provides an instant answer. In particular, comparison of effectiveness of software that uses the differential approach with software that is based upon the integral approach might be very tricky and will strongly depend on a particular application. Therefore, in order to avoid the introduction of misleading information, in this paper we purposely refrain from any speed comparison between our software and the other homology packages.

In Section 2, we introduce the terminology and we carefully define all the notions to be used throughout the remainder of the paper; in particular, we define an AT model and an AM model for a finite cell complex. In Section 3, we provide explicit algorithms for the computation of AT models and AM models. In Section 4, we show how to retrieve selected homological information from the AT models and AM models, such as (co)homology generators. In Section 5, we use this approach to compute the cup product based on formulas at chain level for cubical complexes. In Section 6, we discuss certain additional constructions and variants, like relative (co)homology or reduced (co)homology, which our machinery is capable of handling with very little additional effort. Finally, in Section 7, we briefly describe the prototype software and selected examples that have been made available at the project's website [47], which concludes the paper.

2 Preliminaries

Let R be a Euclidean domain (see e.g. [30], §2.15), that is, a principal ideal domain equipped with the function $\delta_R\colon R \to \mathbb{Z}$ that assigns a non-negative integer to each element of the ring, such that for every $a, b \in R$, $b \neq 0$, there exist $q, r \in R$ for which $a = qb + r$ and $\delta_R(r) < \delta_R(b)$. Intuitively speaking, we assume that R is a commutative ring in which the dividing with remainder is a valid operation. Note that $\delta_R(k) = 1$ if and only if $k \in R$ has its inverse in R, $\delta_R(0) = 0$, and $\delta_R(a) > 1$ for all the non-invertible elements $a \in R$. In the case of the ring or integers $\mathbb{Z}$, one can take $\delta_{\mathbb{Z}}(k) := |k|$. If R is a field, such as the integers modulo a prime number p, denoted

$\mathbb{Z}_p$ (the field frequently used in homology computation, especially for $p = 2$), or the rational numbers, $\mathbb{Q}$, then $\delta_R(k) = 1$ for all $k \neq 0$. An interesting though rarely used in this situation example of a suitable ring might be the ring of polynomials in one variable, with δ_R corresponding to the degree of the polynomial increased by 1 and $\delta_R(0) := 0$.

A *chain complex* (see e.g. [44]) is a graded free abelian group $\{C_q\}_{q\in\mathbb{Z}}$ with a degree -1 homomorphism $\partial_q \colon C_q \to C_{q-1}$ such that $\partial\partial = 0$, called the *boundary operator* or the *differential* of the chain complex. Such a complex arises naturally from a cell complex structure, where each C_q is a free abelian group whose generators correspond to q-dimensional cells, and for each q-dimensional cell c, $\partial_q(c)$ is a linear combination of $(q-1)$-dimensional cells in the boundary of c with the coefficients reflecting incidence numbers and orientation. Thanks to the requirement that $\partial\partial = 0$, the group of *boundaries* $B := \operatorname{im}\partial$ is a subgroup of the group of *cycles* $Z := \ker\partial$, and thus the *homology group* $H := Z/B$ is well defined; since it is an invariant of the topological space (polyhedron) which is the union of all the cells in the complex, it is also called the homology of the topological space. The homology class of a chain $a \in C_q$ is denoted by $[a]$.

Let K be an n-dimensional cell complex, also called a CW complex; see e.g. [27, Ch. 0] for the definition. Denote the set of its q-dimensional cells by $K^{(q)}$. Throughout the entire paper, we shall assume that the number of cells in the complex is finite. The corresponding *chain complex* $(C_q(K), \partial_q)_{q\in\mathbb{Z}}$ over R consists of the free R-modules of q-chains C_q, whose elements are formal combinations of the cells in $K^{(q)}$ with coefficients in R, and a family of homomorphisms $\partial_q \colon C_q \to C_{q-1}$ such that $\partial_{q-1} \circ \partial_q = 0$ and $\partial_q(\sigma)$ is a combination of $(q-1)$-dimensional cells that appear in the boundary of σ with coefficients corresponding to the orientation and multiplicity of these cells (we shall provide explicit formulas in the case of a simplicial complex and a cubical complex below). Note that $C_q(K) = 0$ whenever $q < 0$ or $q > n$. If $R = \mathbb{Z}$ then each C_q is a free abelian group. If R is a field then each C_q is a vector space over R. Since we assume that the number of cells in K is finite, all the R-modules under consideration are finitely generated. On each C_q, we define a bilinear form $C_q \times C_q \ni (c, c') \mapsto \langle c, c'\rangle \in R$ on each pair of generators $\sigma, \tau \in K^{(q)}$ of C_q as follows: $\langle\sigma, \tau\rangle := 1$ if $\sigma = \tau$ and $\langle\sigma, \tau\rangle := 0$ otherwise. This bilinear form is a frequently used formal construct for extracting the coefficient that appears in a chain at a given cell (see e.g. [34]).

As for the actual formula for the boundary operator ∂, the case of a simplicial complex is well established (see e.g. [27]). If a simplex of dimension $q > 0$ is identified by the $(q+1)$-tuple of its (pairwise different) vertices, that is, $\sigma = (v_0, \ldots, v_q)$, then $\partial_q(\sigma) = \sum(-1)^i(v_0, \ldots, \hat{v}_i, \ldots, v_q)$, where the hat over v_i means that v_i is omitted in the q-tuple, and $\partial_q = 0$ if $q \leq 0$ or $q > n$.

The case of a cubical complex is less typical, so we recall some definitions in order to avoid any ambiguity. The reader is referred to [34] for a comprehensive introduction; we also follow some notation from [56]. An *elementary interval* is an interval of the form $[k, k+1]$ (the *non-degenerate* case) or a set $\{k\}$, also denoted as the interval $[k, k]$ or even $[k]$ (the *degenerate* case), where $k \in \mathbb{Z}$. An *elementary cube* in $\mathbb{R}^n$ is the Cartesian product of n elementary intervals, and the number of non-degenerate intervals in this product is its *dimension*.

Let $\sigma = I_1 \times \cdots \times I_n$ be an elementary cube in $\mathbb{R}^n$, where $I_j = \left[a_j^0, a_j^1\right]$ (possibly $a_j^0 = a_j^1$). Let $k_1, \ldots, k_q$ denote those indices that correspond to non-degenerate intervals $I_{k_j} = [a_{k_j}^0, a_{k_j}^1]$ in σ (q is the dimension of σ). For a set $J \subset \{1, \ldots, q\}$, let J' denote the complement of J in $\{1, \ldots, q\}$. Define $k(J) := \{k_i : i \in J\}$, and for $i \in \{0, 1\}$ define the elementary cube $\lambda_J^i \sigma := I_1' \times \cdots \times I_n'$, where $I_j' = \left\{a_j^i\right\}$ if $j \in k(J)$ or $I_j' = I_j$ otherwise. If card $J = 1$, we shall write λ_j^i instead of $\lambda_{\{j\}}^i$. Then define the boundary of σ as follows (see [56], p. 440 in the context of singular cubes):

$$\partial_q(\sigma) := \sum_{j=1}^{q} (-1)^j \left(\lambda_j^0 \sigma - \lambda_j^1 \sigma\right).$$

The homology module of a finite cell complex is a finitely generated module over R, and the classification theorem of finitely generated modules over a p.i.d. (see e.g. [30], §3.9) implies that its description is rather simple: It is a direct sum of its torsion submodule and its free submodule, where the former is a direct sum of primary cyclic modules, and both are finitely generated. Therefore, the homology module $\big(H_q(K)\big)_{q\in\mathbb{Z}}$ is characterized by the ranks of the free submodules for each $H_q(K)$, each called the q-th *Betti number* and denoted $\beta_q(K)$, respectively, and the *torsion coefficients*, which describe the torsion submodules of each $H_q(K)$.

Intuitively speaking, homology groups provide information about different kinds of holes in the space. Namely, β_0 corresponds to the number of connected components (see also the note on reduced homology in Section 6), β_1 is the number of linearly independent holes (like the one inside a circle), β_2 counts the number of voids (like the one surrounded by a sphere), and in general β_q corresponds to the number of defects of the space that look like the hollow space inside the q-dimensional unitary sphere in $\mathbb{R}^{q+1}$. See the figures and discussion in Section 7 for some examples of topological spaces and their homology groups.

Dual concepts lead to the definition of cohomology of a cell complex K, as follows. The *cochain complex* $(C^q(K), \delta^q)_{q\in\mathbb{Z}}$ of K over R consists of homomorphisms from C_q to R, $C^q(K) := \mathrm{Hom}(C_q(K); R)$. The *coboundary operator*, as the dual to ∂, is given by the formula $\big(\delta^q(c)\big)(a) := c\big(\partial_{q+1}(a)\big)$ for $c \in C^q(K)$ and $a \in C_{q+1}(K)$. A cochain $a \in C^q(K)$ is called a *q-cocycle* if $\delta^q(a) = 0$. If $a = \delta^{q-1}(a')$ for some $a' \in C^{q-1}(K)$ then a is called a *q-coboundary*. The q-th *cohomology module* $H^q(K)$ of K is the quotient module of q-cocycles and q-coboundaries. The cohomology class of a cochain $a \in C^q(K)$ is denoted by $[a]$. Although the cohomology groups can be determined from the homology groups using the universal coefficient theorem for cohomology (see e.g. [44], § 53, p. 320), we prefer a direct approach here in order to efficiently use the additional structures necessary in the integral approach, as it will be clear in the sequel. Moreover, note that while chains can be perceived as column vectors containing coefficients of the combinations of cells in K, cochains correspond to row vectors containing coefficients of the dual cochains corresponding to single cells in K (cf. [13]).

If $C_* = \{C_q, \partial_q\}$ and $C'_* = \{C'_q, \partial'_q\}$ are two chain complexes then a *chain map* $f_*\colon C_* \to C'_*$ is a family of homomorphisms $\{f_q\colon C_q \to C'_q\}_{q\in\mathbb{Z}}$ such that $\partial'_q f_q = f_{q-1}\partial_q$. A *chain contraction* from C_* to C'_* is a triple (f, g, ϕ) of chain maps $f\colon C_* \to C'_*$ (*projection*), $g\colon C'_* \to C_*$ (*inclusion*) and $\phi\colon C_* \to C_{*+1}$ (*chain homotopy* or *integral operator*) that satisfy the following conditions:

(a) $\mathrm{Id}_C - gf = \partial\phi + \phi\partial$;
(b) $fg = \mathrm{Id}_{C'}$;
(c) $f\phi = 0$;
(d) $\phi g = 0$;
(e) $\phi\phi = 0$.

This is a classical notion in homological algebra and algebraic topology; see e.g. [16, §12] and comments on the terminology and applications in [22, p. 86]. Note that because of the condition (a), if there exists a chain contraction from C_* to C'_* then their homology and cohomology modules are isomorphic.

A homomorphism $\phi\colon C_* \to C_{*+1}$ is called a *homology gradient vector field* if the following conditions hold:

(a) $\phi\phi = 0$;
(b) $\partial\phi\partial = \partial$;
(c) $\phi\partial\phi = \phi$.

A chain contraction in which ϕ is a homology gradient vector field is called *homology integral chain contraction*.

Definition 1 (**AT model**) An *algebraic-topological model* (introduced in [25]), or an *AT model* for short, of a cell complex K, is a homology integral chain contraction from $C_*(K)$ to some free chain complex M_* with null differential.

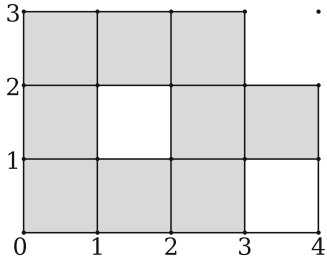

Fig. 1 A sample cubical set K in $\mathbb{R}^2$ consisting of nine squares (indicated in *gray*), two additional 1-dimensional segments ($[3, 4] \times [0]$ and $[4] \times [0, 1]$) and an isolated point ($[4] \times [3]$). Its homology groups over $\mathbb{Z}_2$ are: $H_0 \cong \mathbb{Z}_2 \oplus \mathbb{Z}_2$, $H_1 \cong \mathbb{Z}_2 \oplus \mathbb{Z}_2$, and H_q is trivial for $q \notin \{0, 1\}$. Indeed, the set consists of two connected components and has two holes

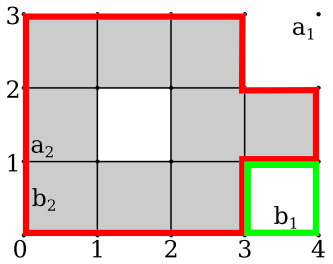

Fig. 2 An AT model (see Definition 1) of the cubical set K shown in Fig. 1 computed with Algorithm 1 (see Section 3). Representants of the four homology generators are: $a_1 = [4] \times [3]$, $a_2 = [0] \times [1]$, $b_1 = [3, 4] \times [0]$ and $b_2 = [0] \times [0, 1]$. These cells correspond to a basis of M_*. Their images by the inclusion g are homology generators of $C(K)$: $g(a_1) = a_1$, $g(a_2) = a_2$, $g(b_1)$ is the loop around the entire gray area, indicated in the figure with a thick line (*red online*), and $g(b_2)$ is the boundary of the missing square in the lower right corner, indicated in the figure with another thick line (*green online*)

The complex M_* in an AT model of K is isomorphic to the homology module of K. An AT model exists for a cell complex if and only if its homology is torsion-free (e.g. if R is a field), and any two AT models of the same complex are isomorphic. Figures 1, 2 and 3 show an example of a cubical complex, its AT model, and a sample application of the AT model to the computation of the coboundary of a homologically trivial cycle. This example is included in the software package available from the project's website [47], and the results shown here were actually obtained by the software.

In fact, given an integral operator ϕ on $C_*(K)$, that is, a homomorphism $\phi\colon C_*(K) \to C_{*+1}(K)$ satisfying $\phi\phi = 0$, it is possible to algebraically determine the other components of the chain contraction (see e.g. [26, 52, 53]); however,

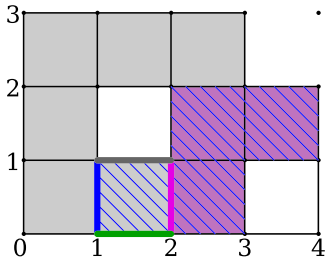

Fig. 3 Sample computation of a coboundary of a homologically trivial cycle, using the integral operator ϕ computed as part of the AT model of K (shown in Figs. 1 and 2). Consider the cycle $c = c_1 + c_2 + c_3 + c_4$, indicated with thick lines in the figure, with $c_1 = [1, 2] \times [0]$ (*green online*), $c_2 = [2] \times [0, 1]$ (*pink online*), $c_3 = [1, 2] \times [1]$ (*gray*), and $c_4 = [1] \times [0, 1]$ (*blue online*). The integral operator applied to these cells is as follows: $\phi(c_1) = 0$, not shown in the figure, $\phi(c_2) = [2, 3] \times [0, 1] + [2, 3] \times [1, 2] + [3, 4] \times [1, 2]$, the three squares shaded in *dark gray (in print)* or in *dark pink (online)*, $\phi(c_3) = 0$, not shown, and $\phi(c_4) = [1, 2] \times [0, 1] + [2, 3] \times [0, 1] + [2, 3] \times [1, 2] + [3, 4] \times [1, 2]$, the four squares indicated with *dashes*. As a consequence, $\phi(c) = c'$, where $c' = [1, 2] \times [0, 1]$. One can check that indeed $\partial c' = c$ (note the field $\mathbb{Z}_2$ of coefficients)

direct construction of M_* and the chain contraction of $C_*(K)$ to M_* (provided in Algorithm 1) makes it easier to obtain explicit representations of homology and cohomology generators (as we show in Section 4).

Definition 2 (**AM model**) An *algebraic minimal model* (introduced in [22]), or an *AM model* for short, of a cell complex K is a chain contraction from $C_*(K)$ to a chain complex (M, d) such that each M_q is a free R-module and all the non-zero elements in the SNF of each d_q are non-invertible in R.

An AM model exists for every cell complex, and any two AM models for the same complex are isomorphic. Moreover, an AM model of a complex K is isomorphic to an AT model of K if the latter exists.

AT models and AM models are the main tools in our approach to computing comprehensive homological information for cell complexes.

3 Algorithms for the computation of chain contractions

Given a cell complex, the following algorithm constructs another complex that represents its homology and computes a homology integral chain contraction of the original cell complex to the new one. It is based on the incremental homology computation algorithm [11, 12], and in its original form was introduced in [25]. Below, we provide this algorithm, rewritten in a detailed way so that the chain maps are defined with respect to the original basis in K and an explicitly constructed set of generators of M_*, which is crucial for effective implementation and for applications.

The input of the algorithm consists of the list of all the cells of a cell complex K ordered as a *filter* $(c_0, \ldots, c_{m-1})$, that is, for each $i = 0, \ldots, m-1$, the set $\{c_0, \ldots, c_i\}$ forms a valid cell complex. In other words, all the cells that appear in the boundary of each cell must precede the cell in the list. A simple filter may be created by sorting the cells in K by their dimension in the ascending order. We assume that a formula for the boundary of each cell is given as a combination of lower-dimensional cells. In order to be consistent with the software implementation, we choose the convention of the C++ programming language to index sequences starting from 0. In order to emphasize the meaning of the components of the chain contraction (f, g, ϕ) being constructed, we denote the projection f by π and the inclusion g by ι in the algorithm. The maps π and ϕ are defined on generators of their domains, and we transparently use them as homomorphisms. The chain complex M_* with the differential $d = 0$ is represented by means of the set of generators $\mathscr{M}$ corresponding to selected cells from K, with their dimension q corresponding to the level q of M_q. We use the subscript index i to indicate the objects constructed at each run of the main loop, but in the software implementation only one instance of these objects exists, and is modified during the computations, so that upon completion of each loop, the previous instance is replaced with the new one.

Algorithm 1 (AT model computation)

INPUT:

$(c_0, \ldots, c_{m-1})$ — a filter of a cell complex K;
∂ — the boundary operator on $C_*(K)$.

PSEUDOCODE:

$\mathscr{M}_{-1} := \emptyset$; $\phi_{-1} := 0$; $\pi_{-1} := \emptyset$; $\iota_{-1} := \emptyset$;
`for` $i := 0$ `to` $m-1$ `do`
  $\phi_i(c_i) := 0$;
  $\bar{c}_i := c_i - \phi_{i-1}(\partial c_i)$;
  `if` $\partial \bar{c}_i = 0$ `then`
    $\mathscr{M}_i := \mathscr{M}_{i-1} \cup \{c_i\}$;
    $\iota_i(c_i) := \bar{c}_i$;
    $\pi_i(c_i) := c_i$;
    `for` $j := 0$ `to` $i-1$ `do`
      $\phi_i(c_j) := \phi_{i-1}(c_j)$;
      $\pi_i(c_j) := \pi_{i-1}(c_j)$;
    `for all` $h \in \mathscr{M}_{i-1}$ `do`
      $\iota_i(h) := \iota_{i-1}(h)$;
  `else`
    `take any` $u_i \in \mathscr{M}_{i-1}$ `such that` $\lambda_i := \langle u_i, \pi_{i-1}(\partial \bar{c}_i)\rangle \neq 0$;
    $\pi_i(c_i) := 0$;
    `for` $j := 0$ `to` $i-1$ `do`
      $\eta_j^i := \langle u_i, \pi_{i-1}(c_j)\rangle$;
      `if` $\eta_j^i \neq 0$ `then`
        $\phi_i(c_j) := \phi_{i-1}(c_j) + \eta_j^i \lambda_i^{-1} \bar{c}_i$;
        $\pi_i(c_j) := \pi_{i-1}(c_j) - \eta_j^i \lambda_i^{-1} \pi_{i-1}(\partial \bar{c}_i)$;
      `else`
        $\phi_i(c_j) := \phi_{i-1}(c_j)$;
        $\pi_i(c_j) := \pi_{i-1}(c_j)$;
      $\mathscr{M}_i := \mathscr{M}_{i-1} \setminus \{u_i\}$;
      $\iota_i := \iota_{i-1}|_{\mathscr{M}_i}$;

OUTPUT:

$\mathscr{M}_{m-1}$ — a set of generators of M_*;
$(\pi_{m-1}, \iota_{m-1}, \phi_{m-1})$ — a chain contraction.

The idea of the algorithm is the following. We process all the cells in the input filter one by one. For each cell c_i, we initially set $\phi(c_i) = 0$, which may be later modified if necessary. If c_i is found to be a cycle then it is added to $\mathscr{M}$, and the values of ι and π are set accordingly. Otherwise, a collapse of c_i through one of its faces u_i is carried out. This means that c_i is not added to $\mathscr{M}$, and u_i is removed from $\mathscr{M}$, with the homomorphisms ϕ and π being modified accordingly to reflect the collapse in which the face u_i is replaced by the remainder of the boundary of c_i.

Note that because of the need to compute the inverse of λ_i, it is necessary to assume the invertibility of all the coefficients encountered in the run of the algorithm at the relevant place; therefore, the algorithm may only be applied safely for field coefficients.

Proposition 1 (see [25]) *In the case of coefficients in a field, Algorithm 1 applied to a filter of a cell complex K returns an AT model of K.*

If the coefficients form a ring which is not a field (e.g. the integers $\mathbb{Z}$) then Algorithm 1 may fail, and thus a more general algorithm must be used. The constructed object is then an AM model. We adapt the algorithm introduced in [22, 26] to obtain a chain contraction and a chain complex (M_*, d) explicitly, with respect to the original basis in $C_*(K)$ that corresponds to the cells in K. This algorithm requires the computation of the Smith Normal Form (SNF) D of the entire boundary matrix ∂, together with the change-of-basis matrix A and its inverse A^{-1}. More specifically, given the matrices of all ∂_q, it is necessary to compute matrices D_q, A_q and A_q^{-1} such that D_q is in the SNF and $D_q = A_{q-1}^{-1} \partial_q A_q$. We say that D is in the SNF and all the entries of D are zero except possibly for $\lambda_{1,1}, \ldots, \lambda_{l,l}$, for some $l \geq 0$, with each $\lambda_{i,i}$ dividing $\lambda_{i+1,i+1}$ for $i = 1, \ldots, l-1$.

The subject of the development of effective algorithms and software for computing the SNF is an intensively investigated area of research on its own. Although the original algorithm given in [58] (see also [44]) has exponential worst-case complexity, as is the case of the naive approach using elementary column and row operations, there exist algorithms for computing the SNF in polynomial time for coefficients in $\mathbb{Z}$ (e.g. [29, 36, 59]), and different variants of these algorithms for solving this problem exist, also crafted to special cases or environments (e.g. for concurrent systems [31]). Unfortunately, these general algorithms turn out to be insufficient for homology computation in practical applications, because of the need to process huge matrices, with hundreds of thousands of rows and columns. However, the boundary matrices derived from cellular complexes are very sparse, that is, have a very small number entries in their columns and rows, which can be used for designing more efficient algorithms (e.g. [15]). Indeed, simple reduction techniques motivated by geometric interpretation of the data may dramatically reduce the matrices in size before coming to a point where no more reductions of this type can be applied and thus the general algorithm for the computation of the SNF must be used. The work by Forman [19] is often used as an inspiration for such reduction techniques. Each geometric reduction corresponds to a change of bases, and thus the matrices A and A^{-1} may be incrementally constructed during the procedure. These reductions correspond e.g. to collapsing external faces or to joining adjacent cells, which were a basis for an algorithm for homology computation by reduction of chain complexes introduced in [33]. Its generalization was implemented in [9] (see also [34]) and is used in the software package available at the project's website [47]. This specific algorithm constructs the SNF matrix D_q of each ∂_q separately, and takes care of keeping generators corresponding to the columns and rows of the already computed SNF at adjacent levels intact, so that, as a result, new bases of all C_q are computed, such that with respect to these bases, all the matrices D_q of ∂_q are in the SNF.

Algorithm 2 (AM model computation)

INPUT:

$(c_0, \ldots, c_{m-1})$ — a filter of a cell complex K;
∂ — the boundary operator on $C_*(K)$.

PSEUDOCODE:

```
compute bases {e_1^q, ..., e_{m_q}^q} of each C_q(K) in which
    the matrices D_q of ∂_q are in the SNF;
    let A_i denote the corresponding
    change-of-basis matrices: D_i = A_{i-1}^{-1} ∂_i A_i;
for q := 0 to dim K do
    for i := 1 to m_q do
        if D_q e_i^q ≠ 0 then
            let j and λ_j^{q-1} be such that D_q e_i^q = λ_j^{q-1} e_j^{q-1};
            if λ_j^{q-1} is invertible then
                ψ^{q-1}(e_j^{q-1}) := λ_j^{q-1} e_i^q;
                continue (take the next i);
            else
                M_{q-1} := M_{q-1} ∪ {m_j^{q-1}};
                d_q(m_i^q) := λ_j^{q-1} m_j^{q-1};
        if there exists k and an invertible λ_i^q
            such that D_{q+1}(e_k^{q+1}) = λ_i^q e_i^q then
            continue (take the next i);
        ψ^q(e_i^q) := 0;
        M_q := M_q ∪ {m_i^q};
        ι(m_i^q) := A_q^{-1}(e_i^q);
        for each j such that γ_j^q := ⟨A_q(c_j), e_i^q⟩ ≠ 0 do
            π(c_j) := π(c_j) + γ_j^q m_i^q;
φ_{q-1} := A_q ψ_{q-1} A_{q-1}^{-1};
```

OUTPUT:

$\mathscr{M}$ — a set of generators of M_*;
d — a differential on M_*;
(π, ι, ϕ) — a chain contraction.

This algorithm is valid in the general case, with coefficients in a Euclidean domain. Note that if $H_*(K)$ has no torsion then this algorithm actually produces an AT model of K, because then the differential of M_* is 0. Moreover, if the coefficient ring is a field then SNF can be computed in cubic time using the simple method based on Gaussian elimination (see e.g. [33]).

The idea of the algorithm is to determine the generators of M_* and the differential d on M_*, as well as the corresponding chain contraction, directly from the SNF of

the boundary operator on K. More specifically, we consider all the elements e_i^1 of a computed collection of bases of $C_q(K)$ in which the boundary matrices are all in the SNF. If the boundary of e_i^q is nonzero then one can deduce that e_i^q is not in the boundary of any e_j^{q+1}, because $D_q D_{q+1} = 0$. In this case, if λ_j^{q-1} is invertible then the pair $\left(e_i^q, e_i^{q-1}\right)$ corresponds to a reduction of the chain complex (a collapse of a cell through its face), and this relation is stored in the chain homotopy operator ψ^{q-1}. If λ_j^{q-1} is non-invertible, on the other hand, then a torsion in (co)homology will arise. Indeed, in this case, e_j^{q-1} is a cycle (note that the boundary of e_j^{q-1} is trivial, because $D_{q-1} D_q = 0$), and at the same time it is not a boundary, while its multiple $\lambda_j^{q-1} e_j^{q-1}$ is a boundary. The projection and inclusion components of the chain contraction can be determined from the matrices A_q and A_q^{-1}, and the chain homotopy operator ψ must be transferred to ϕ in the original bases also using A and A^{-1}.

Proposition 2 (see [22]) *Algorithm 2 applied to a filter of a cell complex K returns an AM model of K. Moreover, for each $e \in \mathscr{M}$, either $d(e) = 0$, or there exists $e' \in \mathscr{M}$ such that $d(e) = \gamma e'$ with a non-invertible $\gamma \in R$.*

An generic implementation in C++ of Algorithms 1 and 2 is the main part of the software package available at the project's website [47], and discussed in Section 7.

4 Homology and cohomology

Let (f, g, ϕ, M_*, d) be an AM model of a cell complex K with the boundary operator denoted by ∂, or an AT model of K if $d = 0$. In this section we shall show how to derive various homological features of K from this model.

The first obvious information to obtain from the AM model of K is a representation of homology of K and cohomology of K. If $d = 0$ (i.e., this is actually an AT model) then $M_* \simeq H_*(K)$, and both Algorithms 1 and 2 provide a set representing a basis of $H_*(K)$, which is a free module. By duality, in this case $H^*(K) \simeq H_*(K)$ and cohomology generators are the duals to homology generators. Otherwise, if $d \neq 0$, some processing of the AM model is necessary; for example, the procedure based on SNF computation of the boundary homomorphisms may be applied to (M_*, d) in order to determine the torsion coefficients of $H_*(K)$ and of $H^*(K)$, as well as the Betti numbers. Fortunately, (M_*, d) is a much smaller object than $(C_*(K), \partial)$, so the expected cost of such computation is negligible, even if a simple (naive) algorithm is applied at this stage. In fact, as stated in Proposition 2, Algorithm 2 actually computes a basis $\mathscr{M}$ for M_*, in which the matrix of d is already in the SNF. Then the set of those $e_q \in \mathscr{M}_q$ ($\mathscr{M}_q$ is a basis of M_q) for which $d_q(e_q) = 0$ corresponds to a set of generators of $H_q(K)$. If there exists $e_{q+1} \in \mathscr{M}_{q+1}$ such that $d_{q+1}(e_{q+1}) = \gamma e_q$ with a non-invertible $\gamma \in R$, then e_q corresponds to a generator of a cyclic submodule of $H_q(K)$ with the torsion coefficient γ; otherwise, e_q generates a free submodule of $H_q(K)$. The number of generators of the latter type is the q-th Betti number of K (see e.g. [44]). Representative cycles that correspond to the homology generators of

K can be instantly obtained by applying the map g to those elements of $\mathscr{M}$ whose boundary is zero, which results in chains in $C(K)$ that are linear combinations of cells in K with coefficients taken from the columns of the matrix representing g in the natural basis in K corresponding to the cells in K and the computed basis $\mathscr{M}$ of M_*.

By duality, each generator $e_q \in \mathscr{M}_q$ of M_q such that $\delta_q e_q^* = 0$ (that is, there is no $e'_{q+1} \in \mathscr{M}_{q+1}$ such that $d_{q+1}\left(e'_{q+1}\right) = \gamma e_q$ with $\gamma \neq 0$) corresponds to a distinct element in a set of generators of $H^q(K)$, and each non-invertible coefficient that appears in $\delta_q e_q^*$ (or, equivalently, in $d_{q+1}\left(e'_{q+1}\right)$) corresponds to a torsion coefficient in $H^{q+1}(K)$, with a generator of the corresponding torsion submodule given by e'_{q+1} such that $d_{q+1}\left(e'_{q+1}\right) = \gamma e_q$. The actual cocycle in $C^*(K)$ that corresponds to a generator e^* of M^* (dual to $e \in \mathscr{M}$) is $e^* \circ f \colon C_*(K) \to R$; the matrix of this map consists of the row corresponding to e in the matrix of f.

If $c \in C_*(K)$ is a cycle then its homology class can be instantly determined by calculating $f(c)$, which provides a linear combination of elements of $\mathscr{M}$. Moreover, if this class is trivial, that is, $f(c) = 0$, then a sample chain $c' \in C_*(K)$ such that $\partial c' = c$ can also be found easily: $c' := \phi(c)$. By duality, if $c \in C^*(K)$ is a cocycle then a representant of its cohomology class is $c \circ g \colon M_* \to R$; the matrix of this map consists of a single row containing the linear combination of elements of the dual basis of M^*.

For $i = 1, 2$, let $(f_i, g_i, \phi_i, M_*^i, d^i)$ be an AM model of the corresponding cell complex K_i with the boundary operator denoted by ∂^i. Using the chain maps g_i to obtain representative cycles of homology classes and f_i to project cycles to homology gives a straightforward way of computing the homomorphism induced in homology by a chain map $h \colon C(K_1) \to C(K_2)$. Namely, for each homology generator of $H_*(K_1)$ represented by $e_q^1 \in \mathscr{M}_q^1$, one can compute its image by $H_*(h)$ as $f_2(h(g_1(e_q^1))) \in M_q^2$, and the matrix of $H_*(h) \colon M_*^1 \to M_*^2$ is the matrix of the homomorphism $f_2 \circ h \circ g_1$. Using the duality, the homomorphism $H^*(h) \colon \mathrm{Hom}(M_*^2; R) \to \mathrm{Hom}(M_*^1; R)$ induced in cohomology by the map h applied to $e \colon M_*^2 \to R$ is given by the formula $H^*(h)(e) = e \circ f_2 \circ h \circ g_1 \colon M_*^1 \to R$, and thus the matrix of $H^*(h)$ is the transpose of the matrix corresponding to $f_2 \circ h \circ g_1$.

The same idea can also be used for the computation of operations that are defined at the chain level and carry over to (co)homology (see [22], Procedure 2): One would take (co)chains that correspond to (co)homology generators, apply the operation to these (co)cycles at the (co)chain level, and then compute the (co)homology class(es) of the resulting (co)cycle(s). This idea was used for computing more advanced (co)homology operations (e.g., Steenrod cohomology operations) in the context of simplicial complexes in [21–23, 50, 51], and we apply it in Section 5 to compute the cubical cup product in cohomology and the cubical version of Alexander-Whitney coproduct in homology.

5 Cubical cup product

The cup product in cohomology is a well known additional structure which transforms the cohomology module of a cell complex into a ring. Although less popular, there is a directly corresponding structure in homology, the Alexander-Whitney coproduct. These operations are defined for simplicial homology by easy and natural formulas at the chain level, but the corresponding formulas for cubical complexes are not that straightforward. Therefore, we devote this section to providing explicit formulas for both operations in the case of cubical complexes, and to discussing various solutions.

The cup product of two simplicial cochains $c \in C^k(K)$ and $c' \in C^l(K)$ is a cochain $c \smile c' \in C^{k+l}(K)$ defined on each generator of C_{k+l} corresponding to a single simplex $\sigma = (v_0, \ldots, v_{k+l})$ as follows:

$$(c \smile c')(\sigma) := c\left(\sigma_0^k\right) \cdot c'\left(\sigma_k^{k+l}\right),$$

where $\sigma_0^k = (v_0, \ldots, v_k)$ and $\sigma_k^{k+l} = (v_k, \ldots, v_{k+l})$, and the dot indicates the multiplication in the ring R. This definition is extended to arbitrary chains by linearity.

The Alexander-Whitney coproduct of a chain consisting of a single n-dimensional simplex $\sigma = (v_0, \ldots, v_n)$ is given by the formula

$$\mathrm{AW}(\sigma) = \sum_{k=0}^{n} \kappa_k^n \sigma_0^k \otimes \sigma_k^n,$$

where $\sigma_i^j = (v_i, \ldots, v_j)$, $\otimes$ is the tensor product over R, and

$$\kappa_k^n = (-1)^0 \cdots (-1)^{k-1} \cdot (-1)^{k+1} \cdots (-1)^n = (-1)^{n(n+1)/2-k}.$$

This formula is extended to arbitrary chains by linearity.

A formula for the cup product of two cubical cochains is more complicated. We follow the idea and the notation from [56, p. 441], where a formula was provided for the multiplication of cubical cochains in the context of *singular* cubical cohomology. If $c \in C^k(K)$ and $c' \in C^l(K)$ are cubical cochains then their cup product is a cubical cochain $c \smile c' \in C^{k+l}(K)$ defined on each generator σ of C_{k+l} corresponding to a single cube (also denoted by σ) as follows:

$$(c \smile c')(\sigma) := \sum_{J \subset \{1,\ldots,k+l\},\ \mathrm{card}\, J = k} \rho_{J,J'}\, c\left(\lambda_{J'}^0 \sigma\right) \cdot c'\left(\lambda_J^1 \sigma\right), \tag{1}$$

where $J' = \{1, \ldots, k+l\} \setminus J$, and $\rho_{J,J'} = (-1)^\nu$, where ν is the number of pairs $(i, j) \in J \times J'$ such that $j < i$. The Alexander-Whitney coproduct of a cube $\sigma \in C^n(K)$ is then given by

$$\mathrm{AW}(\sigma) = \sum_{J \subset \{1,\ldots,n\}} \rho_{J,J'} \left(\lambda_{J'}^0 \sigma \otimes \lambda_J^1 \sigma\right).$$

Let (f, g, ϕ, M_*, d) be an AM model of a cell complex K, or an AT model of K if $d = 0$. Then the cup product of two cochains $c\colon M_i \to R$ and $c'\colon M_j \to R$ is a cochain $c \smile c'\colon M_{i+j} \to R$ given by the following formula:

$$(c \smile c')(\sigma_{i+j}) = (\mu \circ (c \otimes c') \circ (f \otimes f) \circ \mathrm{AW} \circ g)(\sigma_{i+j}),$$

where $\sigma_{i+j} \in M_{i+j}$, $\mu\colon R \times R \to R$ is the multiplication in R, and $\otimes$ denotes the tensor product of homomorphisms defined as follows: $(h_1 \otimes h_2)(x_1 \otimes x_2) = h_1(x_1) \otimes h_2(x_2)$.

An equivalent recursive formula for the cubical cup product (1) was also derived in [35] (see also [32]) in an explicit way in the context of cubical sets as in [34], with [44] as a theoretical basis. The key idea of that construction was to see the cup product in cohomology as a map induced by the composition of two chain maps (see [27, Chapter 3] and [39, Chapter XIII, §3]):

$$C^k(K) \otimes C^l(K) \to C^{k+l}(K \times K) \to C^{k+l}(K),$$

where the first map is the cross product of cochains, defined on the chains corresponding to elementary cubes $Q_1^k \times Q_2^l \subset \mathbb{R}^{k+l}$ by the formula $\left(c_1^k \times c_2^l\right)\left(Q_1^k \times Q_2^l\right) = c_1^k\left(Q_1^k\right) \cdot c_2^l\left(Q_2^l\right)$, and the second map is induced by the diagonal map $K \ni x \mapsto \Delta(x) = (x, x) \in K \times K$. (The majority of [35] is devoted to determining the chain map corresponding to Δ.)

This formula can also be derived by means of simplicial subdivision, using the notion of simplicial sets [40]. A simplicial set is a graded set $X = \{X_k\}_{k \geq 0}$ endowed with two kinds of operators: *face operators* $\partial_i^X\colon X_k \to X_{k-1}$ (for $i = 1, \ldots, k$), and *degeneracy operators* $s_i^X\colon X_k \to X_{k+1}$ (for $i = 0, \ldots, k$); see [40] for the conditions that these operators must satisfy. Elements of X are called *simplices*. In case of "classical" simplices, these two operators are given by the following formulas: $\partial_i^X(v_0, \ldots, v_k) = (v_0, \ldots, v_{i-1}, v_{i+1}, \ldots, v_k)$, and $s_i^X(v_0, \ldots, v_k) = (v_0, \ldots, v_{i-1}, v_i, v_i, v_{i+1}, \ldots, v_k)$. A simplex $x \in X$ is called *degenerate* if $x = s_i^X(y)$ for some i and some $y \in X$. The graded module generated by the degenerate simplices of X is denoted by $s(C_*(X))$. The normalized chain complex $C_*^N(X)$ is the quotient $C_*^N(X) = (C_*(X)/s(C_*(X)), \partial)$, where $\partial = \Sigma(-1)^i \partial_i^X$. Cartesian product $S \times T$ of two simplicial sets is also a simplicial set with $(S \times T)_k = \{(x, y) : x \in S_k,\ y \in T_k\}$, the face operators $\partial_i^{S\times T}(x, y) = (\partial_i^S(x), \partial_i^T(y))$, and the degeneracy operators $s_i^{S\times T}(x, y) = (s_i^S(x), s_i^T(y))$.

An interval $I = [a, b]$ can be obviously considered a simplicial set $S^I = \{S_k^I\}_{k\geq 0}$. An elementary cube $\sigma = I_1 \times \cdots \times I_n$ in $\mathbb{R}^n$, where $I_j = [a_j^0, a_j^1]$ (possibly $a_j^0 = a_j^1$), is the Cartesian product of the intervals, each of which can also be considered a simplicial set. Therefore, the cube σ can be considered both as the simplicial set K^s and a cubical set K^c. The cubical chain complex $C_*(K^c)$ is isomorphic to the tensor product $\bigotimes_j C_*^N(I_j)$. The homological equivalence of the chain complexes $C_*^N(K^s)$ and $C_*(K^c)$ is given by the chain contraction $c_{ez} = (f_{ez}, g_{ez}, \phi_{ez})$ from the chain complex $C_*^N(K^s)$ of the Cartesian product of I_j to the tensor product $C_*(K^c)$. In classical algebraic topology, the chain contraction c_{ez} is called an *Eilenberg-Zilber chain contraction*, and an explicit formulation is given [51, p. 56] (see also [18]). The Alexander-Whitney coproduct $\mathrm{AW}\colon C_*^N(X) \to C_*^N(X) \otimes C_*^N(X)$ is defined on a

simplicial set by $\mathrm{AW}(x) = \partial_0^X \partial_1^X \dots \partial_{i-1}^X(x) \otimes \partial_{i+1}^X \dots \partial_n^X(x)$. The chain contraction c_{ez} can be used to transfer the simplicial Alexander-Whitney coproduct to the cubical context:

$$\mathrm{AW}^c = (f_{ez} \otimes f_{ez}) \circ \mathrm{AW}^s \circ g_{ez}.$$

The Alexander-Whitney coproduct can then be translated to cup product at cochain level by duality:

$$(c \smile c')(\sigma_{i+j}) = (\mu \circ (c \otimes c') \circ \mathrm{AW}^c)(\sigma_{i+j}),$$

where c and c' are cubical cochains of degree i and j, respectively, σ_{i+j} is a cubical chain of degree $i+j$, and $\mu\colon R \times R \to R$ is the multiplication in R.

Instead of delving into further technical details, we shall describe this construction in the case of a 2-dimensional cube. The cubical set $K^c = \{K_0^c, K_1^c, K_2^c\}$ corresponding to the 2-dimensional elementary cube $\sigma = [a,b] \times [c,d]$ is given by

$$\begin{aligned} K_0^c &= \{(a,c),\ (a,d),\ (b,c),\ (b,d)\},\\ K_1^c &= \{[a,b] \times \{c\},\ [a,b] \times \{d\},\ \{a\} \times [c,d],\ \{b\} \times [c,d]\},\\ K_2^c &= \{[a,b] \times [c,d]\}. \end{aligned}$$

The simplicial set $K^s = \{K_0^s, K_1^s, K_2^s\}$ has the following non-degenerate cells:

$$\begin{aligned} K_0^s &= \{(a,c),\ (a,d),\ (b,c),\ (b,d)\},\\ K_1^s &= \{(aa,cd),\ (ab,cc),\ (ab,cd),\ (ab,dd),\ (bb,cd)\},\\ K_2^s &= \{(abb,ccd),\ (aab,cdd)\}. \end{aligned}$$

The chain homotopy operator $\phi_{ez}\colon C_*^N(K^s) \to C_{*+1}^N(K^s)$ helps get rid of the diagonal cell, and is defined as follows: $\phi_{ez}(x) = -(aab,cdd)$ if $x = (ab,cd)$ and $\phi_{ez}(x) = 0$ otherwise. The Alexander-Whitney coproduct of a 2-dimensional simplex is given by

$$\mathrm{AW}^s(w_1,w_2,w_3) = -(w_1) \otimes (w_1,w_2,w_3) + (w_1,w_2) \otimes (w_2,w_3) - (w_1,w_2,w_3) \otimes (w_3).$$

Using c^{ez}, we obtain the following:

$$\begin{aligned} \mathrm{AW}^c([a,b]\times[c,d]) = (\{a\}\times\{c\})\otimes([a,b]\times[c,d]) + ([a,b]\times\{c\})\otimes(\{b\}\times[c,d]) +\\ - (\{a\} \times [c,d]) \otimes ([a,b] \times \{d\}) + ([a,b] \times [c,d]) \otimes (\{b\} \times \{d\}) \end{aligned}$$

We remark that the ring structure introduced by the cup product allows to distinguish some topological spaces which have isomorphic cohomology but different homotopy type. We discuss a few such examples in Section 7. However, the problem of verification whether two rings are isomorphic or not, by means of checking their multiplication tables, is in general a nontrivial one. Therefore, from the practical point of view it is much more efficient to use another invariant which can be derived from the cup product, for example, the homology of the so-called reduced bar construction, which is a classical algebraic invariant in algebraic topology. Since the details are beyond the scope of this paper, we refer the interested readers to [2, 3, 28]. A version of the invariant called HB_1 specific for 3D digital images and derived from this construction was introduced in [24].

6 Additional constructions and variants

Small modifications in the computational machinery introduced above allow for easy implementation of additional useful features, which we discuss in this section. All these features are implemented in the software provided at the project's website [47].

Relative (co)homology If K is a cellular complex and L is its subcomplex (that is, a subset of cells which is a complex itself) then *relative (co)homology* of the pair (K, L) is the (co)homology of the quotient chain complex $C(K)/C(L)$. From the algorithmic point of view, relative (co)homology can be computed as plain (co)homology of the cellular complex $K \setminus L$, in which the cells in L are removed from the boundary of each cell $c \in K \setminus L$.

Reduced (co)homology If we additionally consider the empty cell e as a valid cell of dimension -1, and we put $\partial_0(\sigma) := e$ for every 0-dimensional cell σ, then we obtain the augmented chain complex with $C_{-1} = R$, and its (co)homology is called *reduced (co)homology*. The nice feature of this variant is that the geometric interpretation of reduced homology is more consistent. While the dimension of each homology vector space over a field can be interpreted as the number of "holes" in all dimensions except 0, where it corresponds to the number of connected components of the space, the same interpretation is valid for reduced homology also at level 0, where 0-dimensional "holes" can be interpreted as "gaps" between the connected components. As a consequence, the reduced (co)homology of a "trivial" space, that is, a topological space that is contractible, is zero.

Periodic boundary conditions In various applications, one often imposes periodic boundary conditions in certain directions of a Euclidean space, which corresponds to taking some coordinates from the quotient space $\mathbb{R}/k\mathbb{Z} \simeq S^1$, where $k \in \mathbb{Z}$, $k > 0$. A modification of the definition of cubes and cubical sets for such a space is straightforward, and helps defining certain topological spaces much more easily. For instance, the torus can be viewed as the cubical set $[0, k] \times [0, l]$ in the space $(\mathbb{R}/k\mathbb{Z}) \times (\mathbb{R}/l\mathbb{Z})$, where k, l are arbitrary positive integers.

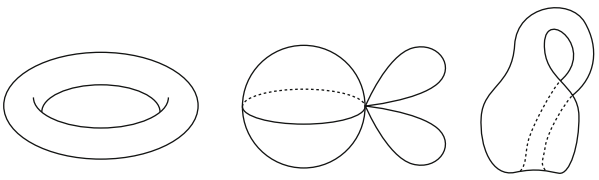

Fig. 4 Three 2-dimensional compact manifolds with the same (co)homology groups over $\mathbb{Z}_2$ but distinguishable using Alexander-Whitney coproduct in homology (see Table 1) or the cup product in cohomology. From *left* to *right*: the torus, the wedge of the sphere and two circles, and the Klein bottle

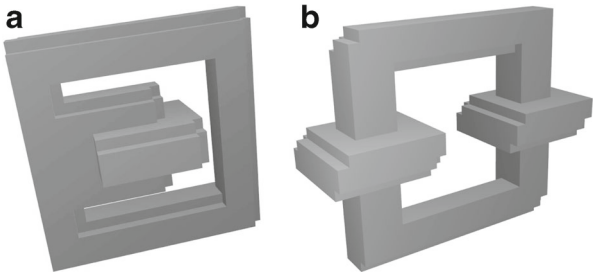

Fig. 5 Cubical approximations of two different configurations of three tori, available in the Voxelo software package, http://munkres.us.es:8080/groups/catam/wiki/e33d2/ (called `Toro A` and `Toro B`). The (co)homology groups of both topological spaces are the same, but the Alexander-Whitney coproduct in homology or the cup product in cohomology can distinguish them (see Table 1)

7 Software and examples

A prototype implementation of Algorithms 1 and 2 together with some illustrative examples are provided at the project's website [47].

The core part of the software is a C++ programming library that defines templates for Algorithms 1 and 2, as well as related data structures and algorithms. The type of cells and the type of coefficients are parameters of the templates, and the algorithms are designed in such a way that they apply to arbitrary types which satisfy the necessary mathematical assumptions. The auxiliary data structures include filtered

Table 1 Results of computations for selected examples

Topological space example	Homology groups over $\mathbb{Z}_2$	Alexander-Whitney coproduct of 2D homology generators, restricted to 1-dimensional chains
Torus	$(\mathbb{Z}_2, \mathbb{Z}_2 \oplus \mathbb{Z}_2, \mathbb{Z}_2)$	$\mathrm{AW}(c_1) = b_1 \otimes b_2 + b_2 \otimes b_1$
$S^2 \wedge S^1 \wedge S^1$	$(\mathbb{Z}_2, \mathbb{Z}_2 \oplus \mathbb{Z}_2, \mathbb{Z}_2)$	$\mathrm{AW}(c_1) = 0$
Klein bottle	$(\mathbb{Z}_2, \mathbb{Z}_2 \oplus \mathbb{Z}_2, \mathbb{Z}_2)$	$\mathrm{AW}(c_1) = b_1 \otimes b_2 + b_2 \otimes b_1 + b_2 \otimes b_2$
Projective plane	$(\mathbb{Z}_2, \mathbb{Z}_2, \mathbb{Z}_2)$	$\mathrm{AW}(c_1) = b_1 \otimes b_1$
$S^2 \wedge S^1$	$(\mathbb{Z}_2, \mathbb{Z}_2, \mathbb{Z}_2)$	$\mathrm{AW}(c_1) = 0$
Three tori (A)	$(\mathbb{Z}_2, \mathbb{Z}_2^4, \mathbb{Z}_2^3)$	$\mathrm{AW}(c_1) = b_3 \otimes b_4 + b_4 \otimes b_3$
		$\mathrm{AW}(c_2) = b_3 \otimes b_4 + b_4 \otimes b_3 + b_1 \otimes b_2 + b_2 \otimes b_1$
		$\mathrm{AW}(c_3) = b_3 \otimes b_4 + b_4 \otimes b_3$
Three tori (B)	$(\mathbb{Z}_2, \mathbb{Z}_2^4, \mathbb{Z}_2^3)$	$\mathrm{AW}(c_1) = b_2 \otimes b_3 + b_3 \otimes b_2 + b_3 \otimes b_4 + b_4 \otimes b_3$
		$\mathrm{AW}(c_2) = b_1 \otimes b_3 + b_3 \otimes b_1 + b_2 \otimes b_3 + b_3 \otimes b_2$
		$\mathrm{AW}(c_3) = b_3 \otimes b_4 + b_4 \otimes b_3$

The torus, the wedge of the sphere and two circles, the Klein bottle, see Fig. 4 (note that these three objects have isomorphic homology over $\mathbb{Z}_2$, but the Alexander-Whitney coproduct allows to distinguish them); the real projective plane, the wedge of the sphere and the circle (again, the same homology, but different Alexander-Whitney coproduct), and two combinations of three tori embedded in $\mathbb{R}^3$ illustrated in Fig. 5 that can be distinguished by the Alexander-Whitney coproduct

complexes, chains, linear maps, and tensor products of chains. A special combinatorial version is provided for $\mathbb{Z}_2$ coefficients; its advantage is that chains correspond to collections of cells, without the need for storing the coefficients. Definitions of cubical cells and simplices are included as sample types of cells, and the rings $\mathbb{Z}$ and $\mathbb{Z}_p$ (where p is a prime number) are also implemented, but other types of cells or rings of coefficients can be also easily added by the users.

In order to simplify using the software, several utility programs are included in the package, which read input data from text files and output the results to the screen in human-readable format. These programs are interfaces to the main features of the C++ software library, and are provided in order to make it easy to try this software; however, using the C++ library interface directly is recommended for intensive applications, as it is much more efficient.

The examples include the well known Klein bottle defined as a simplicial complex, as a surface built of 2-dimensional cubical cells (squares) embedded in $\mathbb{R}^4$, and also as a full cubical set in $\mathbb{R}^4$. There is also the torus and the wedge of a sphere with two circles (see Fig. 4). Note that these three spaces have the same (co)homology groups over $\mathbb{Z}_2$, but can be distinguished with the Alexander-Whitney coproduct (see Table 1) or with the cup product. Another example consists of two different configurations of three tori (see Fig. 5) discussed in [5] (note the wrong result for the Alexander-Whitney coproduct provided there), which also cannot be distinguished by computing (co)homology alone. Results of computations for these and two more examples are gathered in Table 1.

Acknowledgements This research was partially supported from Fundo Europeu de Desenvolvimento Regional (FEDER) through COMPETE – Programa Operacional Factores de Competitividade (POFC) and from the Portuguese national funds through Fundação para a Ciência e a Tecnologia (FCT) in the framework of the research project FCOMP-01-0124-FEDER-010645 (ref. FCT PTDC/MAT/098871/2008), as well as from the funds distributed through the European Science Foundation (ESF) Research Networking Programme on "Applied and Computational Algebraic Topology" (ACAT). P. Real was additionally supported by the Spanish Ministry of Science and Innovation, project no. MTM2009-12716.

References

1. Agoston, M.K.: Algebraic Topology, A First Course, Monographs and Textbooks in Pure and Applied Mathematics, vol. 32. Marcel Dekker, New York (1976)
2. Álvarez, V., Armario, J.A., Frau, M.D., González-Díaz, R., Jiménez, M.J., Real, P., Silva, B.: Computing "small" 1-homological models for commutative differential graded algebras. In: Ganzha, V., Mayr, E., Vorozhtsov, E. (eds.) Computer Algebra in Scientific Computing, pp. 87–100. Springer, Berlin (2000). doi:10.1007/978-3-642-57201-2_9
3. Álvarez, V., Armario, J., Frau, M., Real, P.: Comparison maps for relatively free resolutions. In: Ganzha, V.G., Mayr, E.W., Vorozhtsov, E.V. (eds.) Computer Algebra in Scientific Computing, Lecture Notes in Computer Science, vol. 4194, pp. 1–22. Springer-Berlin, Heidelberg (2006). doi:10.1007/11870814_1
4. Arai, Z., Kalies, W., Kokubu, H., Mischaikow, K., Oka, H., Pilarczyk, P.: A database schema for the analysis of global dynamics of multiparameter systems. SIAM J. Appl. Dyn. Syst. **8**(3), 757–789 (2009). doi:10.1137/080734935
5. Berciano, A., Molina-Abril, H., Pacheco, A., Pilarczyk, P., Real, P.: Decomposing cavities in digital volumes into products of cycles. In: Brlek, S., Reutenauer, C., Provençal, X. (eds.) Discrete Geometry for Computer Imagery, Lecture Notes in Computer Science, vol. 5810, pp. 263–274. Springer-Berlin, Heidelberg (2009). doi:10.1007/978-3-642-04397-0_23

6. Berciano, A., Molina-Abril, H., Real, P.: Searching high order invariants in computer imagery. AAECC **23**(1–2), 17–28 (2012). doi:10.1007/s00200-012-0169-5
7. Bush, J., Gameiro, M., Harker, S., Kokubu, H., Mischaikow, K., Obayashi, I., Pilarczyk, P.: Combinatorial-topological framework for the analysis of global dynamics. Chaos: An Interdisc. J. Nonlinear Sci. **22**(4), 047508 (2012). doi:10.1063/1.4767672
8. Chen, G., Mischaikow, K., Laramee, R.S., Pilarczyk, P., Zhang, E.: Vector field editing and periodic orbit extraction using morse decomposition. IEEE Trans. Vis. Comput. Graph. **13**(4), 769–785 (2007). doi:10.1109/TVCG.2007.1021
9. Computational Homology Project (CHomP) software: http://chomp.rutgers.edu/software/ (2013)
10. Computer Assisted Proofs in Dynamics (CAPD) group: http://capd.ii.uj.edu.pl/ (2013)
11. Delfinado C.J.A., Edelsbrunner H.: An incremental algorithm for Betti numbers of simplicial complexes. In: Proceedings of the Ninth Annual Symposium on Computational Geometry, pp. 232–239. ACM, New York, SCG '93 (1993), doi:10.1145/160985.161140
12. Delfinado, C.J.A., Edelsbrunner, H.: An incremental algorithm for Betti numbers of simplicial complexes on the 3-sphere, grid generation, finite elements, and geometric design. Comput. Aid. Geom. Des. **12**(7), 771–784 (1995). doi:10.1016/0167-8396(95)00016-Y
13. Desbrun, M., Kanso, E., Tong, Y.: Discrete differential forms for computational sciences. In: Discrete Differential Geometry, Course Notes, ACM SIGGRAPH (2006)
14. Dłotko, P., Specogna, R.: Physics inspired algorithms for (co)homology computations of three-dimensional combinatorial manifolds with boundary. Comput. Phys. Commun. **184**(10), 2257–2266 (2013). doi:10.1016/j.cpc.2013.05.006
15. Dumas, J.G., Saunders, B.D., Villard, G.: On efficient sparse integer matrix Smith normal form computations. J. Symb. Comput. **32**(1–2), 71–99 (2001). doi:10.1006/jsco.2001.0451
16. Eilenberg, S., Mac Lane, S.: On the groups $H(\Pi, n)$, I. Ann. Math. **58**, 55–106 (1953)
17. Eilenberg, S., Mac Lane, S.: On the groups $H(\Pi, n)$, II: Methods of computation. Ann. Math. **60**, 49–139 (1954)
18. Eilenberg, S., Zilber, J.: On products of complexes. Am. J. Math. **75**, 200–204 (1953)
19. Forman, R.: Morse theory for cell complexes. Adv. Math. **134**(1), 90–145 (1998). doi:10.1006/aima.1997.1650
20. Gameiro, M., Mischaikow, K., Kalies, W.: Topological characterization of spatial-temporal chaos. Phys. Rev. E **70**, 035,203 (2004)
21. González-Díaz, R., Real, P.: A combinatorial method for computing Steenrod squares. J. Pure Appl. Algebra **139**(1–3), 89–108 (1999). doi:10.1016/S0022-4049(99)00006-7
22. González-Díaz, R., Real, P.: Computation of cohomology operations on finite simplicial complexes. Homology, Homotopy Appl. **5**(2), 83–93 (2003)
23. González-Díaz, R., Real, P.: HPT and cocyclic operations. Homology, Homotopy Appl. **7**(2), 95–108 (2005a)
24. González-Díaz, R., Real, P.: On the cohomology of 3D digital images. Advances in Discrete Geometry and Topology. Discret. Appl. Math. **147**(2–3), 245–263 (2005b). doi:10.1016/j.dam.2004.09.014
25. González-Díaz, R., Medrano, B., Sánchez-Peláez, J., Real, P.: Simplicial perturbation techniques and effective homology. In: Ganzha, V., Mayr, E., Vorozhtsov, E. (eds.) Computer Algebra in Scientific Computing, Lecture Notes in Computer Science, vol. 4194, pp. 166–177. Springer, Berlin (2006). doi:10.1007/11870814_14
26. González-Díaz, R., Jiménez, M., Medrano, B., Real, P.: Chain homotopies for object topological representations. International Conference on Discrete Geometry for Computer Imagery. Discret. Appl. Math. **157**(3), 490–499 (2009). doi:10.1016/j.dam.2008.05.029
27. Hatcher, A.: Algebraic Topology. Cambridge University Press, Cambridge (2002)
28. Hurado, P.R., Álvarez, V., Armario, J.A., González-Díaz, R.: Algorithms in algebraic topology and homological algebra: Problem of complexity. J. Math. Sci. **108**, 1015–1033 (2002). doi:10.1023/A:1013544506151
29. Iliopoulos, C.S.: Worst-case complexity bounds on algorithms for computing the canonical structure of finite Abelian groups and the Hermite and Smith normal forms of an integer matrix. SIAM J. Comput. **18**(4), 658–669 (1989). doi:10.1137/0218045
30. Jacobson, N., 2nd ed. Basic Algebra I. Dover Publications, Mineola (2009)
31. Jäger, G., Wagner, C.: Efficient parallelizations of Hermite and Smith normal form algorithms. Parallel Comput. **35**(6), 345–357 (2009). doi:10.1016/j.parco.2009.01.003

32. Kaczynski, T., Mrozek, M.: The cubical cohomology ring: An algorithmic approach. Foundations of Computational Mathematics (2012). doi:10.1007/s10208-012-9138-4
33. Kaczynski, T., Mrozek, M., Ślusarek, M.: Homology computation by reduction of chain complexes. Comput. Math. Appl. **35**(4), 59–70 (1998). doi:10.1016/S0898-1221(97)00289-7
34. Kaczynski, T., Mischaikow, K., Mrozek, M.: Computational homology, Applied Mathematical Sciences, vol. 157. Springer, New York (2004)
35. Kaczynski, T., Dłotko, P., Mrozek, M.: Computing the cubical cohomology ring. In: González Díaz, R., Real Jurado, P. (eds.): Proceedings of the 3rd International Workshop on Computational Topology in Image Context (CTIC), Research Group on Computational Topology and Applied Mathematics, University of Seville, Seville, Image A, vol. 3, pp. 137–142 (2010)
36. Kannan, R., Bachem, A.: Polynomial algorithms for computing the Smith and Hermite normal forms of an integer matrix. SIAM J. Comput. **8**(4), 499–507 (1979). doi:10.1137/0208040
37. Krishan, K., Gameiro, M., Mischaikow, K., Schatz, M., Kurtuldu, H., Madruga, S.: Homology and symmetry breaking in Rayleigh-Bénard convection: Experiments and simulations. Phys. Fluids **19**, 117–105 (2007)
38. Liz, E., Pilarczyk, P.: Global dynamics in a stage-structured discrete-time population model with harvesting. J. Theor. Biol. **297**, 148–165 (2012). doi:10.1016/j.jtbi.2011.12.012
39. Massey, W.S.: A Basic Course in Algebraic Topology. Addison-Wesley, Menlo Park (1991)
40. May, J.P.: Simplicial objects in algebraic topology, p. 1982. The University of Chicago Press, Chicago (1967). midway reprint
41. Mischaikow, K., Mrozek, M., Pilarczyk, P.: Graph approach to the computation of the homology of continuous maps. Found. Comput. Math. **5**, 199–229 (2005). doi:10.1007/s10208-004-0125-2
42. Molina-Abril, H., Real, P.: Homological spanning forest framework for 2D image analysis. Ann. Math. Artif. Intell. **64**(4), 385–409 (2012). doi:10.1007/s10472-012-9297-7
43. Mrozek, M., Pilarczyk, P., Żelazna, N.: Homology algorithm based on acyclic subspace. Comput. Math. Appl. **55**(11), 2395–2412 (2008). doi:10.1016/j.camwa.2007.08.044
44. Munkres, J.R.: Elements of Algebraic Topology. Addison-Wesley, Reading (1984)
45. Niethammer, M., Stein, A., Kalies, W.D., Pilarczyk, P., Mischaikow, K., Tannenbaum, A.: Analysis of blood vessel topology by cubical homology. In: Proceedings of the International Conference on Image Processing, ICIP 2002, vol. 2, pp. 969–972 (2002). doi:10.1109/ICIP.2002.1040114
46. Niethammer, M., Kalies, W., Mischaikow, K., Tannenbaum, A.: On the detection of simple points in higher dimensions using cubical homology. IEEE Trans. Image Process. **15**(8), 2462–2469 (2006). doi:10.1109/TIP.2006.877309
47. Pilarczyk P.: Chain contractions. Software and examples. http://www.pawelpilarczyk.com/chaincon/ (2013)
48. Pilarczyk, P., Stolot, K.: Excision-preserving cubical approach to the algorithmic computation of the discrete Conley index. Topol. Appl. **155**(10), 1149–1162 (2008). doi:10.1016/j.topol.2008.02.003
49. Pilarczyk, P., García, L., Carreras, B.A., Llerena, I.: A dynamical model for plasma confinement transitions. J. Phys. A Math. Theor. **45**(12), 125,502 (2012). doi:10.1088/1751-8113/45/12/125502
50. Real, P.: On the computability of the Steenrod squares. Ann Univ Ferrara, Nuova Ser, Sez VII Sc Mat **42**, 57–63 (1996). doi:10.1007/BF02955020
51. Real, P.: Homological perturbation theory and associativity. Homol. Homotop. Appl. **2**(5), 51–88 (2000). doi:10.1007/978-3-642-02478-8_52
52. Real, P.: Connectivity forests for homological analysis of digital volumes. In: Cabestany, J., Sandoval, F., Prieto, A., Corchado, J. (eds.) Bio-Inspired Systems: Computational and Ambient Intelligence, Lecture Notes in Computer Science, vol. 5517, pp. 415–423. Springer, Berlin (2009). doi:10.1007/978-3-642-02478-8_52
53. Real, P., Molina-Abril, H.: Cell AT-models for digital volumes. In: Torsello, A., Escolano, F., Brun, L. (eds.) Graph-Based Representations in Pattern Recognition, Lecture Notes in Computer Science, vol. 5534, pp. 314–323. Springer, Berlin (2009). doi:10.1007/978-3-642-02124-4_32
54. Sergeraert, F.: Effective homology, a survey. http://www-fourier.ujf-grenoble.fr/~sergerar/Papers/Survey.pdf (1992)
55. Sergeraert, F.: The computability problem in algebraic topology. Adv. Math. **104**(1), 1–29 (1994). doi:10.1006/aima.1994.1018
56. Serre, J.P.: Homologie singulière des espaces fibrés. Appl. Ann. Math. **54**(3), 425–505 (1951)
57. Shih, W.: Homologie des espaces fibrés. Publ. Math. IHES **13**, 5–87 (1962)

58. Smith, H.J.S.: On systems of linear indeterminate equations and congruences. Philos. Trans. **151**, 293–326 (1861)
59. Storjohann A.: Near optimal algorithms for computing Smith normal forms of integer matrices. In: Proceedings of the 1996 International Symposium on Symbolic and Algebraic Computation, pp. 267–274. ACM, New York, NY, USA, ISSAC '96 (1996). doi:10.1145/236869.237084
60. Veblen, O.: Analysis situs, 2nd ed., Amer. Math. Soc. Colloq. Publ., vol 5, Part 2. Amer. Math. Soc., New York (1931)